

REMARKS

This is intended as a full and complete response to the Final Office Action dated September 12, 2007, having a shortened statutory period for response set to expire on December 12, 2007. Applicants submit this response to place the application in condition for allowance or in better form for appeal. Please reconsider the claims pending in the application for reasons discussed below.

Claims 1-17, 26-41, 50 and 51 are pending in the application. Claims 1-17, 26-41, 50 and 51 remain pending following entry of this response.

Claim Rejections - 35 U.S.C. § 102

Claims 1-17, 26-41, 50-51 are rejected under 35 U.S.C. 102(a) as being anticipated by *Bert Scalzo* ("Oracle DBA Guide to Data Warehousing and Star Schema"), hereinafter "Scalzo." Applicants respectfully traverse this rejection.

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

Regarding claims 1-8, 26-32, and 50:

Applicants submit that Scalzo does not teach a "method of managing execution of query operations in a data processing system" that includes "updating [an] operation status based on a result of [a] subsequent query operation," as recited by claim 1. As claimed, an initial "operation status" is determined for an "initial query operation." Based on the operation status, one of the plurality of subsequent query operations is selected and executed. After executing the subsequent query operation, method of claim 1

includes “updating the operation status based on a result of the subsequent query operation.” Claims 26 and 50 recite similar operations.

The Examiner suggests that *Scalzo*, p. 7, discloses “updating the operation status based on a result of the subsequent query operation.” At page 7, *Scalzo* provides source code for a C program used to illustrate an example of how the “Pro-C language supports “Dynamic SQL.” The specific portions cited by the Examiner include a while loop conditioned on a successful call to “fgets.” In particular, the fgets call sequentially accesses database records, and the while loop processes each one from within the body of the while loop. The Examiner suggests that this process discloses the claimed step of “updating the operation status based on a result of the subsequent query operation. Specifically, the Examiner suggests:

“updating the operation status based on a result of the subsequent query operation” at page 7 (executing “fgets” to update operation status);

Final Office Action, p. 4. However, the “fgets” command does not update the operation status of the subsequent query operation (the “insert” operation - *Scalzo*, page 8). Instead, the “while loop” uses the “fgets” command to simply read in the next line from an input file (*Scalzo*, page 7) until the last line is read. Thus, the operation status of the “insert” statement from the previous loop is not used to select “one of the plurality of subsequent query operations based on the operation status.” However, claim 1 specifically recites “updating the operation status based on a result of the subsequent query operation.” For example, the operation status of an “insert” statement would be updated, and therefore available to use in selecting a subsequent operation before executing remaining query operations. This allows remaining query operations to be managed by the Composite Query Operations Manager (*Specification*, Fig. 3), based upon effects of the previous query operation. For example, “operation status” can provide the number of rows that were inserted (*Specification*, ¶ 57).

Therefore, for all the foregoing reasons, claims 1, 26, and 50 are believed to be allowable, and allowance of the claims is respectfully requested. Furthermore, claims 2-8, and 27-32 depend upon claims 1 and 26, respectively. Accordingly, for all of the

reasons given above, Applicants submit that these dependent claims are allowable and respectfully request allowance of the same.

Regarding claims 9-17, 33-41, and 51:

Applicants submit that *Scalzo* does not teach “managing, using a composite query operations manager, execution of the initial query operation and the plurality of subsequent query operations on the basis of the selection logic and the plurality of failure conditions,” as recited in claims 9, 33, and 51. The Examiner argues that *Scalzo* discloses “managing, using a composite query operations manager, execution of the initial query operation and the plurality of subsequent query operations on the basis of the selection logic and the plurality of failure conditions” at pages 7-8. However, Examiner does not cite any specific subject matter on pages 7-8. Moreover, Applicants submit that the actual content of pages 7-8 does not disclose the claimed steps of managing a plurality of subsequent query operations based on selection logic or failure conditions.

Furthermore, for reasons similar to those set forth above, *Scalzo* does not teach “managing, using a composite query operations manager, execution of the initial query operation and the plurality of subsequent query operations on the basis of the selection logic and the plurality of failure conditions,” as recited in claims 9, 33, and 51. As set forth above, the “fgets” call included in the while loop condition does not update an operation status used to select subsequent query operations. Instead, the “while loop” uses the “fgets” command to simply read in the next line from an input file (*Scalzo*, page 7) until the last line is read. No “operation status” is updated, and therefore, subsequent query operations can not be “managed” based upon previous failure conditions, in the manner claimed. For example, claim 9 recites “managing, using a composite query operations manager, execution of the initial query operation and the plurality of subsequent query operations on the basis of the selection logic and the plurality of failure conditions.” This allows remaining query operations to be managed by the Composite Query Operations Manager (Fig. 3), based upon a plurality of failure conditions. Claims 33 and 51 recite similar limitations.

Therefore, claims 9, 33, and 51 are believed to be allowable, and allowance of the claims is respectfully requested. Furthermore, claims 10-17, and 34-41 depend upon claims 9 and 33, respectively. Accordingly, for all of the reasons given above, Applicants submit that these dependent claims are allowable and respectfully request allowance of the same.

Therefore, the claims are believed to be allowable, and allowance of the claims is respectfully requested.

Conclusion

Having addressed all issues set out in the office action, Applicants respectfully submit that the claims are in condition for allowance and respectfully request that the claims be allowed.

If the Examiner believes any issues remain that prevent this application from going to issue, the Examiner is strongly encouraged to contact Gero McClellan, attorney of record, at (336) 643-3065, to discuss strategies for moving prosecution forward toward allowance.

Respectfully submitted, and
S-signed pursuant to 37 CFR 1.4,

/Gero G. McClellan, Reg. No. 44,227/

Gero G. McClellan
Registration No. 44,227
PATTERSON & SHERIDAN, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Applicants